Chapter 6

Monte Carlo Simulations

These are primarily used to study systems with many degrees of freedom, in particular statistical physics and quantum field theory. In essence, we are using stochastic techniques to evaluate multidimensional integrals.

6.1 Monte Carlo Strategy

Suppose we wish to evaluate the integral

$$I = \int_0^1 f(x)\, dx. \tag{6.1}$$

There are several quadrature methods, such as the trapezoidal rule, which enable us to estimate this, however we could also evaluate I by considering it as the average of the value of f over $x \in [0, 1]$:

$$I \simeq \frac{1}{N} \sum_{i=1}^{N} f(x_i) \tag{6.2}$$

where $\{x_i\}$ are chosen at random with equal probability in the interval $[0, 1]$. The *variance* of this estimate is given by

$$\sigma_I^2 \simeq \frac{1}{N}\sigma_f^2 = \frac{1}{N}\left[\frac{1}{N}\sum_{i=1}^{N} f_i^2 - \left(\frac{1}{N}\sum_{i=1}^{N} f_i\right)^2\right]. \tag{6.3}$$

The error on the integral $\sigma_I \propto 1/N$, whilst the trapezoidal method yields $\sigma_I \propto 1/N^2$. The trapezoidal method is far superior.

The usefulness of this *Monte Carlo* technique lies in the evaluation of *multidimensional* integrals. In a d-dimensional space, for N function evaluations, the Monte Carlo error will still be proportional to $1/N^{1/2}$, whilst for the trapezoidal method $\sigma \propto 1/N^{2/d}$; in general, the Monte Carlo techniques wins for $d > 4$.

In fact, we can do better than this. It is clear from equation (6.3) that the error on the integral would be much smoother were the function f smoother. We will now construct a technique to accomplish this. Let us multiply the denominator and numerator of the integrand by a positive weighting function $w(x)$, normalised so that

$$\int_0^1 w(x)\,dx = 1. \tag{6.4}$$

Then we have

$$I = \int_0^1 dx\, w(x) \frac{f(x)}{w(x)} \tag{6.5}$$

We now introduce a function

$$y(x) = \int_0^x dx'\, w(x') \quad \text{with } y(0) = 0; y(1) = 1. \tag{6.6}$$

Then

$$\frac{dy}{dx} = w(x),$$

and

$$I = \int_0^1 dy \frac{f(x(y))}{w(x(y))}. \tag{6.7}$$

We now compute I using

$$I \simeq \frac{1}{N} \sum_{i=1}^{N} \frac{f(x(y_i))}{w(x(y_i))}, \tag{6.8}$$

where $\{y_i\}$ are chosen randomly with equal probability in the interval $[0, 1]$. Our aim is to choose $w(x)$ so that the ratio f/w is as smooth as possible.

If the distribution in y is *uniform*, the distribution in x is $\frac{dy}{dx} = w(x)$. *The computational effort is concentrated where w, and hopefully f, is large.* This is **importance sampling**.

6.2 Random Number Generators (RNG's)

Monte Carlo methods rely on the ability to generate random numbers with a specified distribution.

The **Probability Distribution** $p(x)$ is defined such that the probability of generating a value between x and $x + dx$ is $p(x)dx$, and is normalised:

$$\int_\infty^\infty p(x)\,dx = 1.$$

Uniform Deviates are the usual starting point for a given probability distribution: all values in a fixed range, usually $[0, 1]$, are equally likely, i.e.

$$p(x)\,dx = \begin{cases} dx & 0 < x < 1 \\ 0 & \text{otherwise} \end{cases} \tag{6.9}$$

A computer is (usually!) a deterministic machine, and therefore we generate "pseudorandom" numbers. The three criteria for a good RNG are:

- There should be a long period, e.g. on a 32-bit machine close to $2^{31}-1$.
- Good "randomness": only small correlations between the numbers generated.
- It should be fast.

The simplest, and most widely used, RNG is the **Linear Congruent Generator**. Here the random numers are generated as an integer sequence $\{I_n\}$ using

$$I_{n+1} = (aI_n + b) \bmod c$$

where a, b, c are integer "magic numbers" whose choice depends on the precision of the machine. A good choice for 32-bit arithmetic is:

$$a = 7^7, b = 0, c = 2^{31} - 1.$$

Then the *uniform deviates* are $x_i = I_i/c$, with maximal period $2^{31} - 1$.

6.2.1 Non-uniform Probability Distributions

Given a uniform deviate $y \in [0, 1]$, we will now see how to construct $x(y)$ such that x is distributed according to $p(x)$.

Transformation Method

The probability that $x(y)$ be less than some value must be the same as that of the uniform deviate being less that y:

$$P_<(x) = \int_{-\infty}^{x} p(x')\,dx' = y, \tag{6.10}$$

or in differential form

$$\frac{dy}{dx} = p(x). \tag{6.11}$$

Example: Exponential distribution $p(x) = e^{-x}, x \geq 0$:

$$P_<(x) = \int_0^x e^{-x'}dx' = 1 - e^{-x} = y, \tag{6.12}$$

yielding

$$x = \ln \frac{1}{1-y}$$

Whilst this method is infallible, for it to be effective, $p(x)$ must be easily invertible.

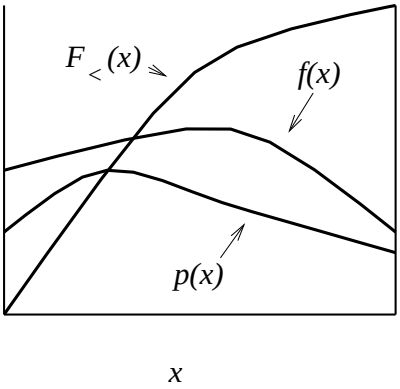

Figure 6.1: The Von Neumann Rejection Method for constructing a probability distribution.

Von Neumann Rejection

This is a method for which we do not have to invert $P_<$ explicitly, or have to evaluate it for a simpler function.

We begin by introducing a function $f(x)$ which is everywhere *greater* than the desired probability distribution $p(x)$, as shown in Figure 6.1. We then construct the incomplete integral

$$F_<(x) = \int^x f(x')dx'; \int_{\text{all x}} f(x')dx' = A. \tag{6.13}$$

Note that $F_<$ should be easily invertible. We then choose a uniform deviate $y \in [0, A]$, and take

$$x = F_<^{-1}(y).$$

We now choose a point ξ uniformly in $[0, f(x)]$. Thus the pair (x, ξ) is distributed *uniformly* in the area under the curve $f(x)$.

Finally, we have the accept/reject step:

- **accept** this x if $\xi \leq p(x)$.
- **reject** this x if $\xi > p(x)$.

The *accepted* values of x are distributed according to $p(x)$.

For this method to be efficient, we require a high acceptance rate, and therefore $f(x)$ should be close to $p(x)$.

6.2.2 Random numbers in higher dimensions

Given $x_i = x_i(y_1, \ldots, y_n), i = 1, \ldots, n$, then joint probability of obtaining $x's$ in interval $[x_1, x_1 + dx_1], \ldots, [x_n, x_n + dx_n]$ is same as probability of obtaining uniform deviates y in interval $[y_1, y_1 + dy_1], \ldots, [y_n, y_n + dy_n]$, i.e.

$$\begin{aligned} p(x_1, \ldots, x_n) dx_1 \ldots dx_n &= dy_1 \ldots dy_n \\ &= \left| \frac{\partial(y_1, \ldots, y_n)}{\partial(x_1, \ldots, x_n)} \right| dx_1 \ldots dx_n \end{aligned} \tag{6.14}$$

Example: Gaussian distribution in two dimensions. We can use

$$\begin{aligned} x_1 &= \sqrt{-2 \ln y_1} \cos 2\pi y_2 \\ x_2 &= \sqrt{-2 \ln y_1} \sin 2\pi y_2 \end{aligned} \tag{6.15}$$

yielding

$$p(x_1, x_2) = \frac{1}{\sqrt{2\pi}} e^{-x_1^2/2} \frac{1}{\sqrt{2\pi}} e^{-x_2^2/2}. \tag{6.16}$$

A more efficient implementation can be constructed using *rejection* method. Pick (v_1, v_2) in unit circle. Then $R = v_1^2 + v_2^2$ is uniform deviate which we can use for y_1, and the angle between (v_1, v_2) and v_1-axis may be used for $2\pi y_2$, yielding

$$\begin{aligned} x_1 &= v_1 \sqrt{\frac{-2 \ln R}{R}} \\ x_2 &= v_2 \sqrt{\frac{-2 \ln R}{R}}. \end{aligned} \tag{6.17}$$

This avoids the need to perform **expensive** trigonometric evaluations.

6.3 Canonical Distribution

The best-known application of importance sampling is to the *canonical distribution.* Here the expectation value of an observable O in equilibrium is

$$\langle O \rangle = \sum_{\mathbf{x}} O(\mathbf{x}) P_{\text{eq}}(\mathbf{x}) \tag{6.18}$$

where

$$P_{\text{eq}}(\mathbf{x}) = \frac{e^{-\beta H(\mathbf{x})}}{\sum_{\mathbf{x}} e^{-\beta H(\mathbf{x})}}. \tag{6.19}$$

is the probability of a given configuration $\mathbf{x} = \{x_n | n = 1, \ldots, N\}$, and $\beta \equiv 1/kT$ is the *inverse temperature.* This could represent;

- Magnetism - x_n is *spin*

- Gauge theories in particle physics - x_n is *quantum field*
- lattice model of an alloy - x_n is *concentration*
- diatomic liquid - x_n is *molecular position/orientation*

Typically, N is very large; we are effectively evaluating an integral of dimension N!

The most straigtforward method would be to pick points $\mathbf{x}$ at random. However, we have seen that this is very inefficient, and therefore once more we employ **importance sampling**. We select configurations $\mathbf{x}_k$ with probability $P(\mathbf{x}_k)$. Then, using equation (6.8), we may write

$$\langle O \rangle \simeq \bar{O} = \frac{\sum_{k=1}^{K} O(\mathbf{x}_k) P^{-1}(\mathbf{x}_k) P_{\rm eq}(\mathbf{x}_k)}{\sum_{k=1}^{K} P^{-1}(\mathbf{x}_k) P_{\rm eq}(\mathbf{x}_k)}. \tag{6.20}$$

The ideal choice would be to employ $P = P_{\rm eq}$ (note that we assume that $O(\mathbf{x}_k)$ is relatively smooth). However, this is rarely feasible.

6.3.1 The Metropolis Algorithm

Metropolis *et al.* [1] (1953) have provided us with a powerful method for sampling points according to the equilibrium distribution $P_{\rm eq}$. A sequence $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t$ is generated as a **random walk** through configuration space in such a way that the probability distribution at the t-th. step satisfies

$$P_t(\mathbf{x}) \stackrel{t\to\infty}{\longrightarrow} P_{\rm eq}(\mathbf{x})\,. \tag{6.21}$$

If equilibrium is reached after $t_{\rm eq}$ sweeps, then

$$\bar{O} = \frac{1}{N} \sum_{k=1}^{N} O(\mathbf{x}_{t_{eq}+k}) \tag{6.22}$$

It is possible to show that the Metropolis Algorithm does indeed produce an equilibrium distribution $P_{\rm eq}$ provided that the sequence $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t$ is generated so that three conditions are met:

1. *normalised*
$$\sum_{\mathbf{x}} W(\mathbf{x} \leftarrow \mathbf{x}') = 1$$
2. *satisfies detailed balance*
$$W(\mathbf{x}' \leftarrow \mathbf{x}) P_{\rm eq}(\mathbf{x}) = W(\mathbf{x} \leftarrow \mathbf{x}') P_{\rm eq}(\mathbf{x}') \; \forall \mathbf{x}, + \mathbf{x}'$$
3. *strongly ergodic*
$$W(\mathbf{x} \leftarrow \mathbf{x}') > 0 \, \forall \mathbf{x}, \mathbf{x}'$$

where $W(\mathbf{x} \leftarrow \mathbf{x}')$ is the transition probability: the probability of stepping from $\mathbf{x}$ to $\mathbf{x}'$ and $P_{\rm eq}(\mathbf{x})$ is the equilibrium distribution of $\mathbf{x}$.

Metropolis Algorithm for Canonical Distribution

Let us look at how the algorithm works.

Starting with a configuration $\mathbf{x}$, we choose a new "trial" configuration $\mathbf{x}'$ with probability $w_{\mathbf{x}'\mathbf{x}} > 0$ satisfying

$$\sum_{\mathbf{x}'} w_{\mathbf{x}'\mathbf{x}} = 1;\ w_{\mathbf{x}'\mathbf{x}} = w_{\mathbf{x}\mathbf{x}'} \text{ (reversibility)}. \tag{6.23}$$

This new configuration is now accepted with probability

$$P_{\text{acc}}(\mathbf{x}' \leftarrow \mathbf{x}) = \min(1, \exp -\beta\Delta H) \tag{6.24}$$

where

$$\Delta H = H(\mathbf{x}') - H(\mathbf{x}). \tag{6.25}$$

If the configuration $\mathbf{x}'$ is *rejected*, then the *new* configuration is $\mathbf{x}$.

The new configuration is accepted if it decreases the energy, and is accepted with probability $e^{-\beta\Delta H}$ if it increases the energy.

Thus the transition probability is

$$W(\mathbf{x}' \leftarrow \mathbf{x})|_{\mathbf{x}'\neq\mathbf{x}} = w_{\mathbf{x}'\mathbf{x}} P_{\text{acc}}(\mathbf{x}' \leftarrow \mathbf{x}) \tag{6.26}$$

and

$$\begin{aligned} W(\mathbf{x} \leftarrow \mathbf{x}) &= w_{\mathbf{x}\mathbf{x}} + \sum_{\mathbf{x}'\neq\mathbf{x}} w_{\mathbf{x}'\mathbf{x}}(1 - P_{\text{acc}}(\mathbf{x}' \leftarrow \mathbf{x})) && (6.27)\\ &= 1 - \sum_{\mathbf{x}'\neq\mathbf{x}} w_{\mathbf{x}'\mathbf{x}} P_{\text{acc}}(\mathbf{x}' \leftarrow \mathbf{x}), && (6.28)\end{aligned}$$

where the second term on the r.h.s of equation (6.27) represents the probability that the new configuration is *rejected*

This satisfies the conditions (1)-(3) above for convergence to the canonical distribution:

1. *Normalisation*: from the above, we have

$$\sum_{\mathbf{x}'} W(\mathbf{x}' \leftarrow \mathbf{x}) = W(\mathbf{x} \leftarrow \mathbf{x}) + \sum_{\mathbf{x}'\neq\mathbf{x}} W(\mathbf{x}' \leftarrow \mathbf{x}) = 1$$

2. *Detailed balance*: for $\mathbf{x}' \neq \mathbf{x}$, we have

$$\begin{aligned} \frac{W(\mathbf{x}' \leftarrow \mathbf{x})}{W(\mathbf{x} \leftarrow \mathbf{x}')} &= \frac{w_{\mathbf{x}'\mathbf{x}} P_{\text{acc}}(\mathbf{x}' \leftarrow \mathbf{x})}{w_{\mathbf{x}\mathbf{x}'} P_{\text{acc}}(\mathbf{x} \leftarrow \mathbf{x}')}\\ &= \frac{\min(1, e^{-\beta(H(\mathbf{x}')-H(\mathbf{x}))})}{\min(1, e^{-\beta(H(\mathbf{x})-H(\mathbf{x}'))})} \text{ since } w_{\mathbf{x}\mathbf{x}'} = w_{\mathbf{x}'\mathbf{x}}\\ &= e^{-\beta(H(\mathbf{x}')-H(\mathbf{x}))}\\ &= \frac{P_{\text{eq}}(\mathbf{x}')}{P_{\text{eq}}(\mathbf{x})}\end{aligned}$$

3. *Ergodicity*: from equations (6.26) and (6.28) we have

$$W(\mathbf{x}' \leftarrow \mathbf{x}) > 0 \, \forall \, \mathbf{x}', \mathbf{x}.$$

Implementation:

We now describe a practical implementation on a *serial* machine. We assume that each configuration $\mathbf{x}$ is specified by N dynamical variables $(x_1, \ldots, x_N)$. Then our procedure is as follows:

1. Pick a starting configuration $\mathbf{x}_0$. Typical examples are when each x_i is random (*hot start*) or set to a constant value (*cold start*).

2. Go through the variables in order, $i = 1, \ldots, N$, updating each in turn keeping the others fixed. Thus for $i = 1, \ldots, N$:

   (a) Pick a new trial value x_i' at random with probability $w_{x'x}$ satisfying (6.23).

   (b) Calculate the change in energy $\Delta H = H(x_i') - H(x_i)$, and compute $P_{\rm acc}(x_i' \leftarrow x_i) = e^{-\beta \Delta H}$.

   (c) Choose a random deviate $r \in [0, 1]$:

$$\begin{aligned} r \leq P_{\rm acc}(x_i' \leftarrow x_i) \qquad & \text{accept the change, } x_i \rightarrow x_i' \\ r > P_{\rm acc}(x_i' \leftarrow x_i) \qquad & \text{reject the change, } x_i \rightarrow x_i \end{aligned}$$

3. One complete pass through the variables is a *sweep*. Return to (2).

- We cannot obtain measurements of quantities until we have performed sufficient sweeps that the system is in equilibrium; in general, local quantities equilibrate *faster* than long-range correlations.

- Configurations at the end of successive sweeps are **correlated**. If the cost of making measurements is high, then it is more economical to make measurements on configurations separated by sufficient sweeps that they are uncorrelated, e.g.

$$\bar{O} = \frac{1}{K} \sum_{k=1}^{K} O(\mathbf{x}_{t_{eq} + k.t_{corr}})$$

  wehre $t_{\rm corr}$ is the separation between our "uncorrelated" configurations. Once again, the choice of $t_{\rm corr}$ depends on the O, and is larger near a phase transition.

In most problems in statistical physics, H is predominantly *local*, and for each x_n the accept/reject step involves nearby data.

It is often straightforward to derive a *data parallel* implementation of this algorithm. In particular, it is possible to update in parallel those dynamical variables not connected through H. Thus in the case of the Ising model, we can use the *red-black decomposition* introduced earlier.

6.3.2 Statistical Errors

Let us now look briefly at the **errors** which affect Monte Carlo simulations. We begin with the following assumptions

1. The system has reached equilibrium, $t > t_{\rm eq}$
2. Successive measurements are made on **uncorrelated** configurations; we will return to the discussion of *autocorrelations* later.

Distribution mean and variance

Suppose we sample a continuous random variable y with probability distribution $p(y)$. Then the **mean** of y^n is

$$\langle y^n \rangle = \int y^n p(y)\, dy, \tag{6.29}$$

and the **variance** of y is

$$\begin{aligned} \sigma^2 &\equiv \langle (y - \langle y \rangle)^2 \rangle \\ &= \langle y^2 \rangle - \langle y \rangle^2 \end{aligned} \tag{6.30}$$

Sample mean and variance

We make N measurements y_i, distributed according to $p(y)$. The the **sample mean** of y^n is

$$\overline{y^n} = \frac{1}{N} \sum_{i=1}^{N} y_i^n. \tag{6.31}$$

We now want to show that the expectation value of the *sample mean* is the expectation value of the *true mean*. We thus consider repeating our experiment of making N measurement of y many times:

$$\begin{aligned} \langle \overline{y^n} \rangle &= \langle \frac{1}{N} \sum_{i=1}^{N} y_i^n \rangle \\ &= \frac{1}{N} \sum_{i=1}^{N} \langle y_i^n \rangle \\ &= \frac{1}{N} \times N \langle y^n \rangle = \langle y^n \rangle. \end{aligned} \tag{6.32}$$

In particular, we say that $\overline{y}$ is an **unbiased estimator** of $\langle y \rangle$.

We will now study the *variance*. First, we note

$$\begin{aligned} \langle \overline{y}^2 \rangle &= \langle \left(\frac{1}{N} \sum_{i=1}^{N} y_i \right)^2 \rangle \\ &= \frac{1}{N^2} \sum_{i,j=1}^{N} \langle y_i y_j \rangle \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{N^2}\sum_{i=1}^{N}\langle y_i^2\rangle + \frac{1}{N^2}\sum_{i\neq j}^{N}\langle y_i y_j\rangle \\
&= \frac{1}{N}\langle y^2\rangle + \frac{N-1}{N}\langle y\rangle^2 \qquad (6.33)
\end{aligned}
$$

where we have used that the y_i are *uncorrelated*.

The **sample variance** is defined by

$$\overline{\sigma}^2 \equiv \frac{N}{N-1}\langle \overline{y^2} - \overline{y}^2\rangle. \qquad (6.34)$$

Then we have

$$
\begin{aligned}
\overline{\sigma}^2 &= \frac{N}{N-1}\left(\langle y^2\rangle - \frac{1}{N}\langle y^2\rangle - \frac{N-1}{N}\langle y\rangle^2\right) \\
&= \frac{N}{N-1}\left(\frac{N-1}{N}\langle y^2\rangle - \frac{N-1}{N}\langle y\rangle^2\right) \\
&= \sigma^2, \qquad (6.35)
\end{aligned}
$$

so that the *sample variance* is an *unbiased estimator* of the *true variance*.

In our simulations we will measure an observable to obtain the mean $\overline{y}$. We require the variance of $\overline{y}$:

$$
\begin{aligned}
\sigma_{\overline{y}}^2 &= \langle \overline{y}^2\rangle - \langle \overline{y}\rangle^2 \\
&= \frac{1}{N}\langle y^2\rangle + \frac{N-1}{N}\langle y\rangle^2 - \langle y\rangle^2 \\
&= \frac{1}{N}(\langle y^2\rangle - \langle y\rangle^2) \\
&= \frac{1}{N}\sigma^2. \qquad (6.36)
\end{aligned}
$$

The estimates for the **mean value** and **statistical error** are

$$\overline{y} \pm \frac{\overline{\sigma}}{\sqrt{N}}, \qquad (6.37)$$

where $\overline{\sigma}$ is given in equation (6.34).

Autocorrelations

Correlations between measurements made on different configurations complicate the analysis, and our expression for $\sigma_{\overline{y}}^2$, considerably. In particular, we will *overestimate* the statistical errors on our measurements. The simplest way to correctly estimate the errors is to *bin* successive blocks of data of length B. In particular, we form

$$\overline{y}_b = \frac{1}{B}\sum_{j=1}^{B} y_{b.B+j}, \quad b = 0, \ldots, \frac{N}{B} - 1. \qquad (6.38)$$

We then repeat the analysis above as if we had made N/B measurements, using $\overline{y}_b$ in place of y_i. The **variance of the sample mean** will then **increase** with **increasing** B until successive blocks are **uncorrelated**, and a true estimate of the error is obtained.

6.3.3 Systematic Uncertainties

As well as the *statistical errors* discussed above, there are various sources of **systematic uncertainties**, arising from the "systematics" of the simulation that cannot be decreased by increasing statistics.

Finite-size Effects:

We model a large physical system, e.g. magnet, by the simulation of a smaller system with many fewer sites. If the system is characterised by a correlation length ξ, then effects at the boundary are likely to be important when ξ approaches L, the length of our simulation "box".

Finite size scaling exploits this dependence on system size to extrapolate measurements to infinite volume.

Finite-size Scaling Hypothesis: *the behaviour of a system with linear size L near the infinite-volume critical temperature, T_c, depends only on L/ξ where ξ is the correlation length of the infinite system.*

Critical Slowing Down:

The error on the sample mean is proportional to $\sqrt{\tau}$, where τ is the autocorrelation time. In the neighbourhood of a continuous phase transition, τ diverges with a **dynamical critical exponent** z:

$$\tau \propto \xi^z \propto |T - T_c|^{-\nu z} \tag{6.39}$$

where the correlation length diverges at critical temperature T_c as

$$\xi \propto |T - T_c|^{-\nu}.$$

This is *critical slowing down.*

In a box of linear size L, the maximum correlation length is L so that, when $\xi > L$, the autocorrelation time satisfies

$$\tau \propto L^z,$$

and thus, for a d-dimensional system, the computation time increases as

$$\text{computation} \propto L^{d+z},$$

i.e. the *computation increases faster than the memory*, and the need is to find algorithms for which z is a minimum.

Discretisation Effects:
We often are modelling continuous variables by variables on a discretised grid. This introduces systematic uncertainties, for example in replacing derivatives by the difference operator.

6.4 Hybrid Monte Carlo

Consider simulating an atomic gas of N atoms in a fixed volume by Metropolis Monte Carlo. Suppose the system has reached equilibrium. At each step, we attempt a random displacement of each atom and accept the change with probability

$$\min\left\{1, e^{-\Delta H}\right\} \tag{6.40}$$

where ΔH is the change in energy.

If the displacement is large, the energy change may be large and then the acceptance probability will be small.

If the displacement is small, so that the acceptance probability is high, progress through configuration space will be slow, almost a random walk in which $O(N^2)$ steps are required to reach a distance which could be reached in N steps in a more orderly fashion.

Large steps with high acceptance probability lie in the energy surface $\Delta H = 0$. These are defined by Newton's Laws of Motion, which are the basis of **Molecular Dynamics (MD)**.

Consider a system of N particles interacting via pairwise potentials with Hamiltonian

$$H(\mathbf{x}, \mathbf{p}) = \sum_{j=1}^{N} \frac{\mathbf{p}_j^2}{2m_j} + \sum_{j>k} V(r_{jk}), \qquad r_{jk} = |\mathbf{x}_j - \mathbf{x}_k|. \tag{6.41}$$

The equations of motion are

$$\dot{\mathbf{p}}_i = -\nabla_i \sum_{j>k} V(r_{jk}) = \mathbf{F}_i \tag{6.42}$$

$$\dot{\mathbf{x}}_i = \frac{\mathbf{p}_i}{m_i} \tag{6.43}$$

where $\mathbf{F}_i$ is the force on particle i due to all the other particles.

In a **MD simulation**, one starts from some initial configuration and integrates *approximately* the equations of motion, conserving the total energy, E. The system is equilibrated by rescaling the momenta until the average kinetic energy is $kT/2$ per degree of freedom. If the dynamics is **ergodic**, then the configurations generated sample the **microcanonical ensemble**:

$$P_\mu(\mathbf{x}, \mathbf{p})\; d^N\mathbf{x}\, d^N\mathbf{p} = \frac{1}{Z_\mu}\, \delta\left(H(\mathbf{x}, \mathbf{p}) - E\right)\; d^N\mathbf{x}\, d^N\mathbf{p}. \tag{6.44}$$

In a **MC simulation**, one again starts from an arbitrary set of coordinates and momenta and generates a Markov chain of configurations using a Metropolis update procedure. The update involves picking a **trial** configuration, $(\mathbf{x}', \mathbf{p}')$, by adding small random increments to the current configuration, $(\mathbf{x}, \mathbf{p})$, with a probability which is the same as for the reverse process. Then the new configuration is accepted with probability

$$P_{\rm acc} = \min\left(1, e^{-\beta\Delta H}\right) \tag{6.45}$$

where $\beta = 1/kT$ and $\Delta H = H(\mathbf{x}', \mathbf{p}') - H(\mathbf{x}, \mathbf{p})$. This procedure satisfies **detailed balance**:

$$d^N\mathbf{x}\, d^N\mathbf{p}\; e^{-\beta H(\mathbf{x},\mathbf{p})} \min\left(1, e^{-\beta\Delta H}\right) = d^N\mathbf{x}'\, d^N\mathbf{p}'\; e^{-\beta H(\mathbf{x}',\mathbf{p}')} \min\left(1, e^{\beta\Delta H}\right) \tag{6.46}$$

and so the probability distribution for the generated configurations converges to the **canonical distribution**:

$$P_{eq}(\mathbf{x}, \mathbf{p})\; d^N\mathbf{x}\, d^N\mathbf{p} = \frac{e^{-\beta H(\mathbf{x},\mathbf{p})}}{Z}\; d^N\mathbf{x}\, d^N\mathbf{p} \tag{6.47}$$

in which the system is in equilibrium with a heatbath at temperature T.

Note: in the thermodynamic limit, $N \to \infty$, $V \to \infty$, $N/V =$ constant, the microcanonical and canonical ensembles are equivalent, and so MC and MD simulations should give the same results for equilibrium quantities provided the ergodic hypothesis holds.

However, if the trial configuration involves changing *too many* degrees of freedom at once, or if *too large a change* is made in a few degrees of freedom, then the new configuration is likely to have a small Boltzmann probability and the trial configuration will usually be rejected. On the other hand, small changes evolve the system very slowly, in something like a random walk through configuration space, and the procedure is inefficient.

Hybrid Monte Carlo (HMC) is based on MD, but has **no discretisation errors** and **reaches equilibrium more rapidly** than Metropolis MC.

A single **HMC update** consists of the following steps:

1. Replace all momenta with new values, $\mathbf{p}_i$, drawn from the correct equilibrium distribution

$$P(\mathbf{p}_i) \propto \exp\left(-\beta\frac{\mathbf{p}_i^2}{2m_i}\right). \tag{6.48}$$

   This is easily done by generating a set of Gaussian random numbers with unit variance and rescaling them appropriately.

2. With probability one half, flip the signs of *all* momenta.

3. Employing a *time-reversible* discretisation scheme, integrate the equations of motion over a fixed time period, t_0, to obtain a new set of coordinates $\mathbf{x}_i'$ and momenta $\mathbf{p}_i'$.

4. Accept the new (primed) configuration with Metropolis probability, P_{acc}, given by Eq (6.45). If the new configuration is rejected, restore the old $\mathbf{x}_i$ and $\mathbf{p}_i$.

The first step trivially generates the desired distribution for the momenta.

The procedure satisfies detailed balance, Eq (6.46), because the probability of picking $(\mathbf{x}', \mathbf{p}')$ starting from $(\mathbf{x}, \mathbf{p})$ is the same as that of picking $(\mathbf{x}, \mathbf{p})$ starting from $(\mathbf{x}', \mathbf{p}')$. It is $1/2$ in each case, because flipping the momenta, $\mathbf{p}_i$, is equivalent to integrating the equations of motion *backwards* in time.

In practice, we do not need to flip the signs of the momenta, because step 1 generates the new momenta with a probability which depends only on the magnitude of $\mathbf{p}_i$; the sign is already random.

The leapfrog algorithm provides a time-reversible integration scheme:

$$\begin{aligned}
\mathbf{p}_i(t+\delta t/2) &= \mathbf{p}_i(t) + \dot{\mathbf{p}}_i(t)\,\delta t/2 = \mathbf{p}_i(t) + \mathbf{F}_i(t)\,\delta t/2 \\
\mathbf{x}_i(t+\delta t) &= \mathbf{x}_i(t) + \dot{\mathbf{x}}_i(t+\delta t/2)\,\delta t = \mathbf{x}_i(t) + \frac{\mathbf{p}_i(t+\delta t/2)}{m_i}\,\delta t \qquad (6.49) \\
\mathbf{p}_i(t+\delta t) &= \mathbf{p}_i(t+\delta t/2) + \dot{\mathbf{p}}_i(t+\delta t)\,\delta t/2 = \mathbf{p}_i(t+\delta t/2) + \mathbf{F}_i(t+\delta t)\,\delta t/2.
\end{aligned}$$

It also satisfies **Liouville's Theorem:** the measure $d^N\mathbf{x}\,d^N\mathbf{p}$ is preserved during the timestep. This is necessary for HMC to satisfy the detailed balance relation, Eq (6.46).

The error in $\mathbf{x}$ in one timestep is $O(\delta t^3)$. So, if one performs $t_0/\delta t$ iterations before performing the Metropolis accept/reject step, then the accumulated error is $O(\delta t^2)$.

In the limit $\delta t \to 0$, the equations of motion are **integrated exactly**, energy is conserved, and $\Delta H = 0$, so **every trial configuration is accepted**. This is the strength of HMC: provided the equations of motion are integrated sufficiently accurately, even large changes in the configurations are accepted.

Note that the Metropolis accept/reject step **removes all discretisation errors** associated with the leapfrog procedure, and the generated configurations occur with the Boltzmann probability independent of the value of δt.

Analytic calculations suggest that ΔH depends on the combination $N\delta t^4$, so the cost of a calculation, at fixed acceptance rate, grows as $N^{\frac{5}{4}}$.

Reversible measure-preserving discretisation schemes, which are accurate to arbitrary order in δt, have been constructed. A simple example of

a scheme which is accurate to $O(\delta t^5)$ consists of three successive iterations of equations (6.49). The first and third iterations have a stepsize δt, whilst the second has stepsize $-2^{\frac{1}{3}}\delta t$. The $O(\delta t^3)$ errors cancel between the backward and forward steps. Iteration of this procedure produces results to any desired order in δt.

Reference

1. N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller and E. Teller, "Equation-of-State Calculations by Fast Computing Machines", Journ. of Chem. Phys. **21** (1953) 1087.